DICIEMBRE DE 2018

AUTOMATAS CELULARES Y EL JUEGO DE LA VIDA

MATEMÁTICAS PARA LA COMPUTACIÓN

JAVIER ORTIZ PÉREZ – 32093767G
FABIO RODRÍGUEZ MACÍAS – 75892789B
INGENERÍA INFORMÁTICA – TECNOLOGÍAS INFORMÁTICAS
Universidad de Sevilla

Resumen

En este proyecto vamos a realizar una documentación acerca de los autómatas celulares. Hablaremos sobre su historia, sus diferentes tipos y las aplicaciones que tienen en la vida real. A parte de esto desarrollaremos el Juego de la Vida, donde podremos ver a este autómata gráficamente funcionando.

Agradecimientos

A nuestros profesores de Matemáticas para la Computación y a Miguel Ángel Gutiérrez Naranjo del departamento de CCIA por su inspiración.

Índice

Contenido

o

Capítulo 1: Autómatas celulares

1.1 Historia.

El origen de los autómatas celulares se produce en los años 50, siendo como principal precursor uno de los mayores matemáticos de la era moderna, John Von Neumann, mientras intentaba crear un modelo de autoreproducción en biología y sin el uso de ordenadores.

Ilustración 1 John Von Neumann

No fue hasta finales de los años 50 cuando se consiguieron unos resultados mucho más detallados y técnicos que probaban la capacidad computacional de los autómatas celulares siendo capaces de, en determinadas situaciones, funcionar con la misma capacidad de una máquina de Turing.

En 1968, aun habiendo poca investigación, John Conway empezó a trabajar en una gran variedad de reglas para autómatas celulares de dos dimensiones. Fue en 1970 cuando expuso una serie de reglas a las que llamó “El Juego de la Vida” y el cual ha sido uno de los autómatas celulares más famosos. Estas reglas a pesar de ser muy simples exhibían comportamientos complejos. Fue un juego de gran interés para matemáticos, físicos e incluso economistas.

En la actualidad los Autómatas Celulares son una importante rama de las matemáticas y la computación y son un tema muy interesante de estudio. Tienen gran cantidad de aplicaciones en diversas ramas de la ciencia como en la física, biología…

1.2 Descripción.

1.2.1 Definición

La Wikipedia define autómata celular como:
"Un autómata celular (A.C.) es un modelo matemático para un sistema dinámico que evoluciona en pasos discretos. Es adecuado para modelar sistemas naturales que puedan ser descritos como una colección masiva de objetos simples que interactúen localmente unos con otros. "

Lo más importante de esta definición es que es un modelo matemático que evoluciona y que tiene una colección de objetos simples. Uno de los aspectos que más caracteriza a los AC es la capacidad de lograr algo a través del paso del tiempo y no desde un inicio. Esto hace que no sea fácil analizarlos si no es a través de una simulación, partiendo de un estado o configuración inicial de células.

1.2.2 Elementos de los autómatas celulares

Es imposible ser capaz de explicar en qué consiste un Autómata celular sin hacer mención de sus elementos básicos:

- Rejilla: Formada por enteros (conjunto $\mathbb{Z}$) que se extiende de forma infinita. A cada celda de la cuadrícula se le conocerá como **célula**. Esta puede ser de un plano de 2 dimensiones a un espacio n-dimensional.
- Conjunto de estados: es finito y cada una de las células toma un valor de este conjunto de estados. Se puede expresar con valores o colores.
- Configuración inicial: consiste en darle un determinado valor a cada una de las células del espacio para que a partir de esta configuración comiencen a evolucionar.
- Vecindad: define el conjunto de células que rodean a una célula determinada.
- Función local: regla o reglas de evolución que define como evolucionará el autómata. Definen como cambia una célula teniendo en cuenta el valor de esta y de sus vecinos.

1.2.3 Condiciones de la frontera

Para realizar y entender mejor la representación visual que se usa en los Autómatas Celulares, es necesario definir los tipos de límite o fronteras en los espacios donde se desarrolla el algoritmo.

- Frontera abierta: se considera que todas las células fuera de la rejilla toman un valor fijo.
- Frontera reflectora: las células fuera del tablero toman los valores que están dentro, como si fuera un espejo.
- Frontera periódica o circular (2): se considera el tablero como si sus extremos se tocarán, es decir, como si realizáramos un pliegue en forma de cilindro.
- Sin frontera: la representación del autómata no tiene límites, es infinito, a medida que se va extendiendo y llega a los límites representaríamos una mayor cantidad de células en pantalla como si le fuéramos quitando zoom a la simulación. Obviamente esto tiene un límite ya sea impuesto por el programador o por límite de memoria.

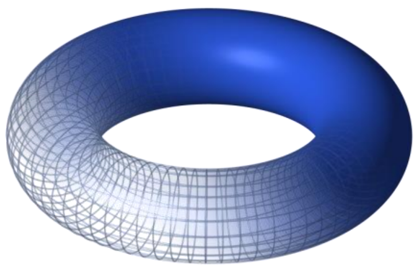


Ilustración 2: Autómata Celular con frontera periódica

Digamos que estas condiciones son las básicas que se suelen aplicar a la creación de los autómatas celulares, pero esto no debe de ser necesariamente así pudiéndole añadir cualquier modificación que se crea necesaria o interesante.

1.2.4 Variaciones

Como antes hemos mencionado todas las características nombradas hasta ahora pueden variar dando lugar a autómatas celulares *no estándar*. Un ejemplo de esto podría ser en vez de representar las células con cuadrados representarlas con cualquier otra forma geométrica por ejemplo un triángulo.

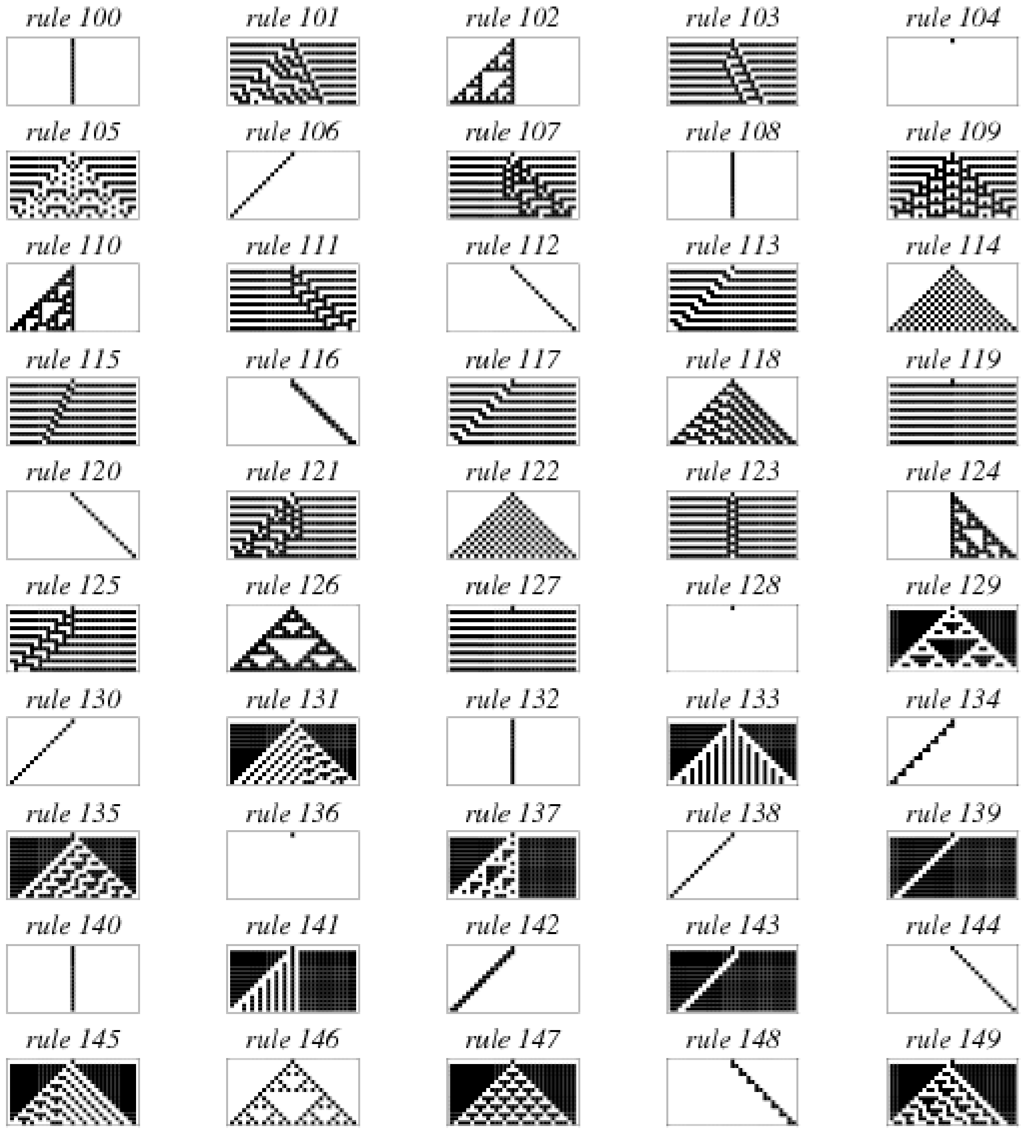
1.2.5 Clasificación de los Autómatas Celulares.

Stephen Wolfram empezó a trabajar en autómatas celulares a mediados del 1981. Sus investigaciones fueron impulsadas por un interés en sistemas modelados como las redes neuronales. Al ver la complejidad del comportamiento de estas reglas simples Wolfram llego a sospechar que la complejidad de la naturaleza puede ser debida a mecanismos similares. En 2002 Wolfram publicó un texto de 1280 páginas, A New Kind of Science, donde explica que los autómatas celulares tienen una gran importancia para todas las disciplinas de la ciencia.

Wolfram define cuatro clases en las que se pueden clasificar los Autómatas Celulares.

- **Clase I:** La configuración inicial llega a una configuración estable homogénea (todas las células con el mismo valor).
- **Clase II:** La evolución lleva al patrón inicial a un conjunto de estructuras simples que son estables o periódicas. Parte de la aleatoriedad del patrón inicial puede permanecer.
- **Clase III**: La configuración evoluciona de forma caótica.
- **Clase IV**: La evolución lleva a estructuras complejas e interesantes con la formación de estructuras locales capaces de sobrevivir por largos periodos de tipo. El resultado final puede ser un Clase 2 pero el número de pasos requeridos para llegar a este estado puede ser uy grande incluso cuando el patrón inicial es relativamente simple.

1.2.6 Reglas de los autómatas celulares



Existe una gran cantidad de reglas con las que somos capaces de dibujar una gran cantidad de patrones. Sin ir más lejos, la regla 90, es capaz de crear el fractal "Triángulo de Sierpinski" estudiado en la asignatura.

1.2.7 Aplicaciones de los autómatas celulares.

Aunque a simple vista los autómatas celulares pueden parecer un modelado muy simple, tienen una gran cantidad de aplicaciones en la vida real. Hay que tener en cuenta que, si somos capaces de relacionar un sistema real conceptos de “vecindad”, “estado de los componentes” y “función de transición” hay altas probabilidades de que se modelable a través de un Autómata celular.

Ejemplos aplicaciones y similitudes con sistemas reales son:

- Crecimiento de algunos cristales como los patrones en copos de nieves, capaces de ser modelizados por A.C en dos dimensiones.
- Patrones encontrados en moluscos Conus Textile (ilustración 3) son similares a los generados por autómatas celulares de una dimensión con reglas determinadas.
- Uso en criptografía, si se usan reglas suficientemente caóticas, se pueden cifrar cadenas las cuales serán altamente costosas de descifrar si no se conocen las reglas.
- Modelizaciones de fluidos.
- Modelizaciones de flujos de tráficos.

Ilustración 3 Caparazón de Conus Textile

1.4 Otro ejemplo de Autómata Celular.

Hormiga de Langton:

La hormiga de Langton es una máquina de Turing bidimensional con un conjunto de reglas muy sencillo que sin embargo da lugar a comportamientos complejos. Fue inventada por Chris Langton en 1986 y su universidad la demostró en el año 2002.

Sus reglas son:

1. Si está sobre un cuadrado blanco, cambia el color del cuadrado, gira noventa grados a la izquierda y avanza un cuadrado.
2. Si está sobre un cuadrado negro, cambia el color del cuadrado, gira noventa grados a la derecha y avanza un cuadrado.

En su comportamiento podemos ver:

Simplicidad: durante los primeros cientos de pasos la hormiga crea patrones sencillos y frecuentemente simétricos.

Caos: después aparece un patrón irregular y grande. La hormiga sigue un camino aparentemente al azar.

Orden emergente: finalmente la hormiga empieza a construir un "camino" que consistirá en un patrón de 104 casos que se repetirá indefinidamente.

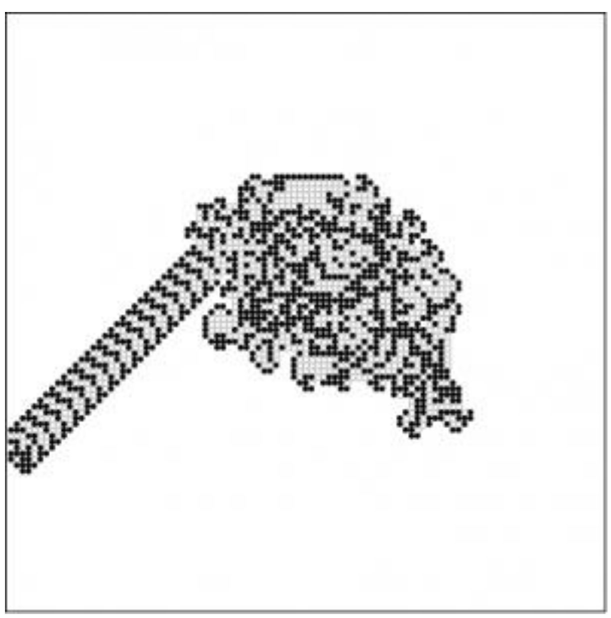


Ilustración 4 Hormiga de Langton

1.5 Relación de los A.C con el contenido de la teoría

Relación de los autómatas celulares con los vistos en teoría. Lo que hemos visto en teoría son **sistemas dinámicos discretos**, **teoría del caos** y **fractales**. Antes de continuar debemos recordar la definición de sistemas dinámicos y de autómata celular.

- Sistemas dinámicos discretos: sistema cuyo estado evoluciona con el tiempo. Este se mide en pequeños lapsos.
- Autómata celular: modelo matemático para un sistema dinámico que evoluciona en pasos discretos.

Viendo las dos definiciones es bastante claro que un autómata celular un sistema dinámico discreto. Con diferentes modelizaciones seremos capaces de que el autómata realice un objetivo concreto. Lo que es tan interesante de esto es que, si somos capaces de encontrar un sistema real que interactúan localmente ente si y podamos analogar conceptos de vecindad, estados y función de transición, podremos ser capaces de generar un sistema dinámico y conseguir simular este a través de un Autómata Celular. Un ejemplo de simulaciones de modelos físicos reales que hoy se estudian con autómatas celulares son modelado de fluidos, flujo de tráfico, evolución de células o virus como el VIH…

Un sistema dinámico discreto evoluciona en el tiempo de tal forma que:

$$X_k = X_{k+1}$$

Por ejemplo, si estamos estudiando cómo evoluciona la reproducción de los peces, podemos pensar que X_k serán la cantidad de peces que existen en la generación k y X_{k+1} el número de peces que existen en la siguiente generación (años, por ejemplo). Es decir, sabiendo el número de peces que existe en una generación determinada somos capaces de saber cómo evolucionará y cuantos habrá en el siguiente.

Esta explicación de cómo funciona un sistema dinámico discreto puede ser llevada a los autómatas celulares que funcionan exactamente igual, evolucionando a lo largo del tiempo. Es por esto por lo que son capaces de simular, si sabemos lo suficientemente bien cómo funcionan, sistemas físicos.

Ejemplos de esta modelización podría ser modelado de fluidos o modelado de evolución de virus como VIH. Obviamente estos campos están en actualmente en estudio y el grado de efectividad que se obtiene no es del 100%. También hay que tener en cuenta que la modelización de un sistema real en reglas no es tarea sencilla y no siempre es realizable

Relacionando los AC con los sistemas dinámicos estudiados en la asignatura, en los AC **cada casilla** evoluciona de la forma que lo hacen los sistemas dinámicos discretos de una generación a otra con una función que este caso devolverá un resultado u otro dependiendo del estado de los vecinos de la célula que está evolucionando. Es por esto por lo que los autómatas celulares son buenos en la simulación de sistemas, ya que podemos representar gráficamente como trabajara haciendo su estudio mucho más visual. En el caso del juego de la Vida la función sería:

$$f(V_k) = K_{n+1}$$

$Donde\ V\ son\ los\ vecinos\ de\ la\ célula\ k$ y n es la generación.

Es decir, para una determinada célula, si a la función de evolución del sistema le introducimos el valor de sus células vecinas nos devolverá su valor en la generación siguiente. Un ejemplo interesante de aplicación real y que usaremos para explicar todo esto es un autómata celular que realice una predicción de un incendio. A través de imágenes teledetectadas por satélite somo capaces de realizar una clasificación del terreno. Es decir, indicar en cada píxel si está compuesto de árbol, tierra, agua… Será esto lo que usaremos como tablero para el A.C.

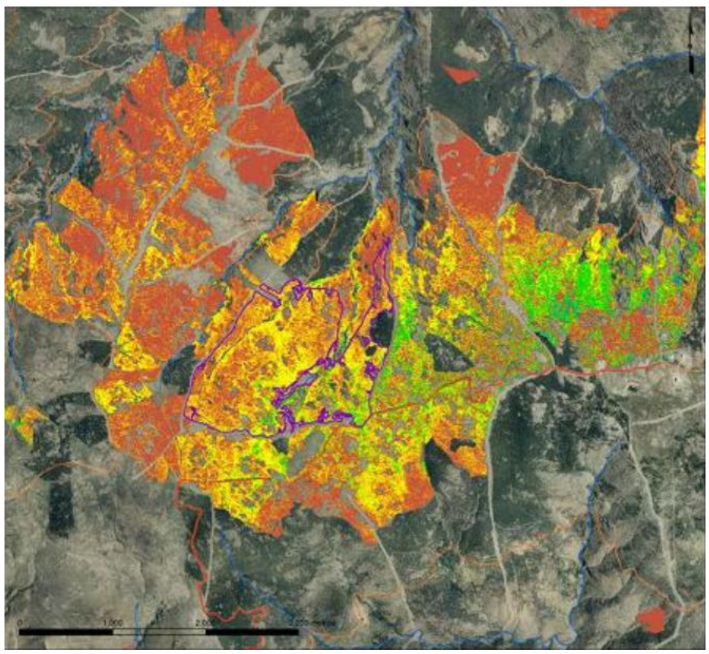


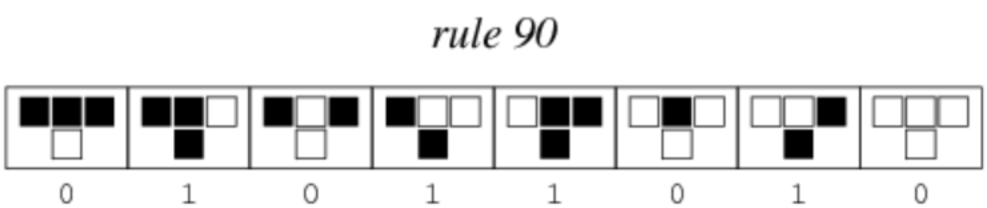
Ilustración 5 Clasificación de imagen de satélite

Si somos capaces de modelar matemáticamente la acción del viento en el fuego y cómo evoluciona en un entorno de bosque (algo que se hace hoy en día) seremos capaces de realizar la simulación a través de un A.C.

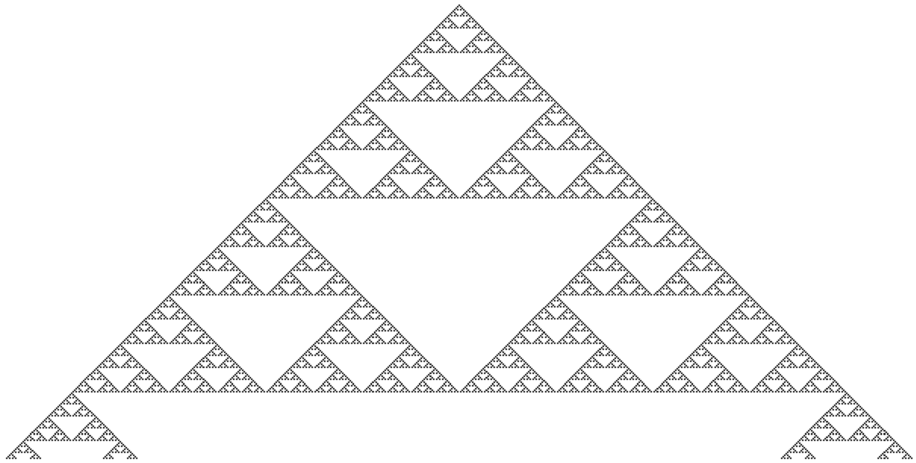
En la asignatura también hemos estudiado los fractales. Estos los define Wikipedia como: "Un **fractal** es un objeto geométrico cuya estructura básica, fragmentada o aparentemente irregular, se repite a diferentes escalas". Los fractales son objetos autosimilares, esto es que su forma se hace realizando formas más pequeñas de la misma figura. En la naturaleza aparece infinidad de veces generando figuras que vemos todos los días en la naturaleza, por ejemplo, en el brécol romanesco o las ramas de los árboles. Es por esto mayormente que es un interesante tema de estudio.

Ilustración 6 Geometría del brécol

Existe una gran cantidad de fractales que son capaces de reproducirse usando diferentes reglas de autómatas celulares. Un ejemplo de esto es el triángulo de Sierpinski, estudiado en la asignatura, que es capaz de formarse usando la regla 90.



Lo cual genera el fractal siguiente “Triángulo de Sierpinski:



Capítulo 2: El juego de la vida

1.1 Introducción

El Juego de la Vida es una aplicación de autómatas celulares ideada por John Horton Conway en 1970. Es el mejor ejemplo de autómatas celulares conocido hasta ahora.

La evolución del juego es determinada por un estado inicial, sin la necesidad de interacción humana, aunque es posible interactuar con el juego creando una configuración inicial, observando posteriormente como evoluciona.

1.2 Reglas

El universo del juego de la vida es una cuadrícula ortogonal infinita de celdas cuadradas en dos dimensiones, donde cada celda tiene dos estados posibles: viva o muerta. Cada celda interactúa con sus ocho *vecinos*, los cuales son las celdas colindantes a la dada. En cada paso en el tiempo, ocurren las siguientes transiciones:

1. Cualquier célula viva con menos de dos vecinos vivos muere (subpoblación).
2. Cualquier célula con más de tres vecinos vivos muere (superpoblación).
3. Cualquier célula con dos o tres vecinos vivos vive.
4. Cualquier célula muerta con exactamente tres vecinos vivos vuelve a la vida.

El patrón inicial constituye la *semilla* del sistema. La primera generación se crea mediante la aplicación simultanea de las reglas anteriores a cada célula de la semilla. El nacimiento y la muerte de las células ocurre simultáneamente, llamándole *tick* a este momento discreto. Las reglas se siguen aplicando continuamente creando nuevas generaciones de células.

1.3 Orígenes

Conway estuvo interesado en el problema presentado en la década de los 40 por el renombrado matemático John von Neumann, quien intento buscar una máquina hipotética que pudiera construir copias de sí mismo. Tuvo éxito cuando encontró un modelo matemático para una máquina con reglas complicadas en una cuadrícula rectangular. El Juego de la Vida surgió del intento de Conway de simplificar las ideas de von Neumann. Hizo su primera aparición pública en la edición de octubre de los 70 de *Scientific American*, en la columna de *juegos matemáticos* de Martin Gardner, bajo el título *The fantastic combinations of John Conway's new solitaire game "life"*. Desde un punto de vista teórico, es interesante porque tiene el poder de una máquina de Turing universal: es decir, cualquier cosa que pueda calcularse algorítmicamente puede calcularse dentro del juego de la vida de Conway. Gardner escribió:

> *El juego hizo a Conway famoso inmediatamente, pero también abrió un nuevo campo de investigación matemática, el campo de los autómatas celulares debido a sus analogías con organismos vivos, pertenece a una clase creciente de lo que se llaman juegos de simulación.*

Desde su publicación, el Juego de la Vida ha atraído mucho interés debido a las sorprendentes formas en que los patrones pueden evolucionar. Es interesante para físicos, matemáticos, biólogos, economistas, filósofos y todo tipo de científicos observar la forma en la que pueden surgir patrones complejos de la implementación de reglas muy simples. El juego también puede servir como una analogía didáctica, utilizada para transmitir una noción contradictoria a la intuición de que diseño y organización pueden surgir espontáneamente en ausencia de un diseñador.

La popularidad del Juego de la Vida se vio favorecida por su creación para una nueva generación de minicomputadoras baratas que se lanzaron al mercado, lo que significa que el juego podría ejecutarse durante horas en estas máquinas. Para muchos, el Juego de la Vida era simplemente un desafío de programación, una forma divertida de perder ciclos de CPU. Para otros, sin embargo, tenía connotaciones filosóficas. Desarrolló un culto a lo largo de los años 70 y más allá. Los desarrollos actuales han llegado tan lejos como para crear emulaciones teóricas de sistemas informáticos dentro de los límites de un tablero.

Conway eligió sus reglas cuidadosamente, después de una considerable experimentación, para cumplir tres criterios:

1. No debe haber un patrón inicial para el cual haya una prueba simple de que la población puede crecer sin límite.
2. Debería haber patrones iniciales que aparentemente crecen sin límite.
3. Debe haber patrones iniciales simples que crezcan y cambien durante un periodo de tiempo considerable antes de llegar a su fin de las siguientes maneras posibles: desapareciendo completamente o estableciéndose en una configuración estable que permanece sin cambios a partir de entonces, o entrando en una fase oscilante en la que repiten un ciclo sin fin de dos o más periodos.

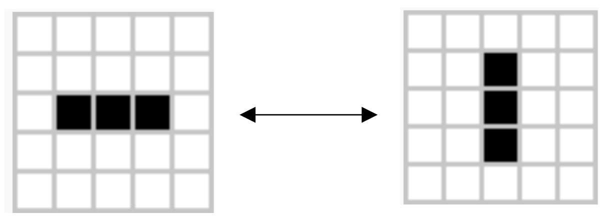
1.4 Patrones

En el juego de la vida existen diferentes patrones de células.

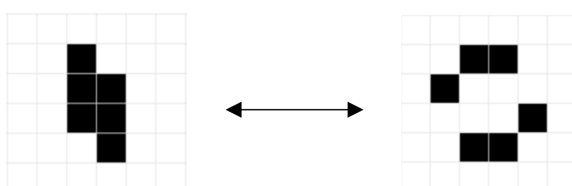
OSCILADORES

Tras un número finito finito de oscilaciones vuelven a su estado inicial. El número de generaciones define el periodo del oscilador. Se han descubierto osciladores de todos los periodos, ya que hay reglas para generar osciladores de cualquier periodo deseado.

Los osciladores cuentan con un rotor, que son las células que cambian de estado en algún momento de la evolución del oscilador, y con un estator, que son las células que permanecen vivas durante todas las fases de la evolución del oscilador.



Blinker (periodo 2)



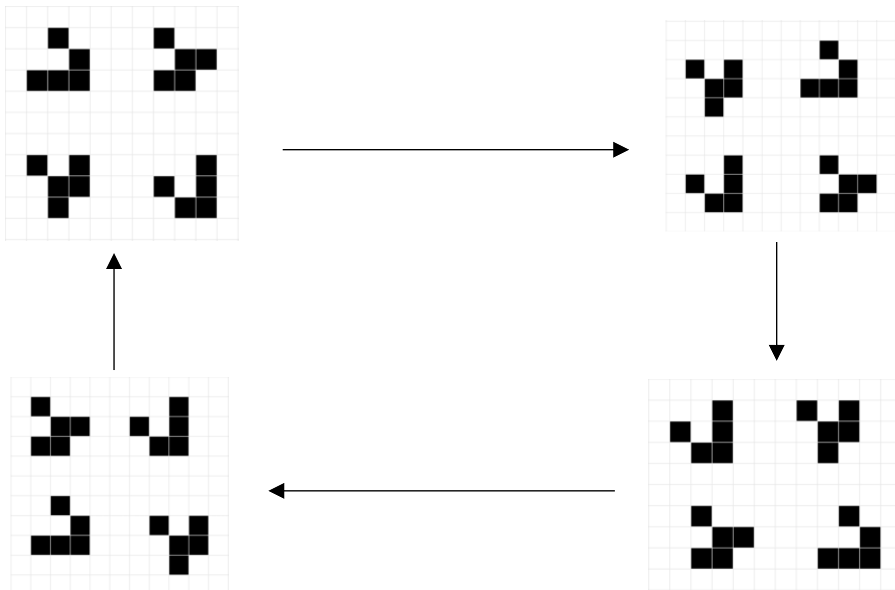
Toad (periodo 2)

VIDAS ESTÁTICAS

Son patrones que no cambian de una generación a otra. Se consideran osciladores de periodo 1 y se asume que son estáticas y no vacías.

NAVES ESPACIALES

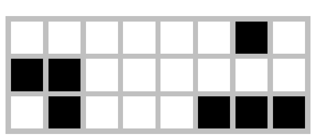
Son patrones que tras un número finito de generaciones vuelven a su estado original pero en una ubicación diferente. La velocidad de una nave es el número de celdas que se desplaza dividido por la longitud de su período. El máximo posible es una celda por generación, velocidad que se conoce como *c*.



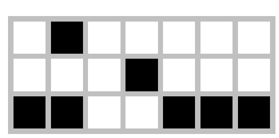
Spaceship (periodo 4)

MATUSALENES

Los matusalenes son patrones que pueden evolucionar a lo largo de muchos turnos, o generaciones, antes de estabilizarse. El patrón Diehard desaparece después de 130 ticks, mientras que Acorn tarda 5206 ticks en estabilizarse en forma de muchos osciladores, y en ese tiempo genera 13 planeadores.



Diehard



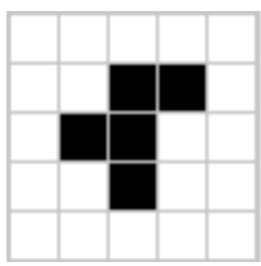
Acorn

1.5 Iteraciones

A partir de un patrón inicial aleatorio de células vivas en la cuadrícula, lo encontraremos un cambio constante en la población a medida que pasan generaciones. Los patrones que emergen de las reglas simples pueden ser considerados como una forma de belleza. Los pequeños subpatrones aislados sin simetría inicial tienden a volverse simétricos. Una vez que esto sucede, la simetría puede aumentar en riqueza, pero no puede perderse a menos que un subpatrón cercano se acerque lo suficiente como para perturbarla. En muy pocos casos, la sociedad finalmente se extingue y todas las células vivas desaparecen. La mayoría de los patrones iniciales se "queman", produciendo figuras estables o patrones que oscilan para siempre entre dos o más estados (conocidos como cenizas); muchos también producen uno o más planeadores o naves espaciales que viajan indefinidamente lejos de la ubicación inicial.

1.6 Algoritmos

Los primeros resultados en el Juego de la Vida se obtuvieron sin el uso de computadoras. Las naturalezas muertas y los osciladores más simples se descubrieron mientras se rastreaban los destinos de pequeñas configuraciones iniciales utilizando papel cuadriculado, pizarras, tableros de juego físicos (como Go) y similares. Durante esta investigación inicial, Conway descubrió que el R-pentomino no se estabilizó en un pequeño número de generaciones.



R-pentomino

Estos descubrimientos inspiraron a los programadores de todo el mundo a escribir programas para rastrear la evolución de los patrones de Vida. La mayoría de los algoritmos iniciales eran similares. Representaban los patrones de vida como matrices bidimensionales en la memoria de la computadora. Normalmente, se utilizan dos matrices, una para mantener la generación actual y otra para calcular su sucesor. A menudo, 0 y 1 representan células muertas y vivas, respectivamente. Un bucle doble considera cada elemento de la matriz actual a su vez, contando los vecinos vivos de cada celda para decidir si el elemento correspondiente de la matriz sucesora debe ser 0 o 1. Se muestra la matriz sucesora. Para la siguiente iteración, los *arrays* intercambian roles para que la matriz sucesora en la última iteración se convierta en la matriz actual en la siguiente iteración.

Se garantiza que, si una celda no cambió en el último *tick*, y ninguno de sus vecinos cambiaron, tampoco cambiará en el siguiente *tick*, por lo que un programa que permita realizar un seguimiento de qué áreas están activas puede ahorrar tiempo al no actualizar zonas inactivas.

En principio, el tablero del Juego de la Vida es infinito, pero las computadoras tienen memoria finita y, por lo general, los tamaños de los *arrays* deben declararse por adelantado. Esto conduce a problemas cuando el área activa invade el borde de la matriz. Los programadores han utilizado varias estrategias para abordar estos problemas. La estrategia más sencilla es asumir que cada celda fuera de la matriz está muerta. Esto es fácil de programar, pero conduce a resultados inexactos cuando el área activa cruza el límite. Una estrategia más sofisticada es considerar la unión de sus bordes formando una matriz toroidal. El resultado es que las áreas activas que se mueven a través del borde de un campo reaparecen en el borde opuesto. La inexactitud aún puede resultar si el patrón crece demasiado, pero al menos no existe inexactitud en los bordes.

Alternativamente, el programador puede representar el tablero con un tipo de estructura de datos diferente, como un vector de pares de coordenadas que representan las células vivas. Este enfoque permite que el patrón se mueva por el campo sin obstáculos. El inconveniente es que contar vecinos se convierte en una operación de búsqueda, lo que ralentiza la velocidad de simulación. Con estructuras de datos más complejas, se podría resolver el inconveniente.

Para explorar grandes patrones en gran profundidad de tiempo, se usan algoritmos sofisticados como *Hashlife*, que es un algoritmo creado por Bill Gosper en 1984 para simular el Juego de la Vida. La idea general del algoritmo es almacenar subpatrones en una tabla *hash* para que los resultados de su evolución no tengan que volver a calcularse si vuelven a aparecer en otro lugar o tiempo. Se almacenan y reutilizan sin tener que volver a calcularlos, siempre que vuelvan a aparecer los mismos.

1.7 Variaciones *(Life-like cellular automaton)*

Desde la creación original del juego, se han desarrollado nuevas reglas basadas en ideas similares. El juego de la vida estándar, se simboliza como *B3/S23*. El primer conjunto de números, indicado por “B” para el nacimiento, es la lista de números vecinos que una célula muerta puede tener para nacer en la próxima generación. El segundo conjunto, indicado por “S”, para la supervivencia, es lo que se requiere para que una célula viva continúe. Por lo tanto, “B6/S16” significa “una célula nace si hay 6 vecinos, y vive si hay 1 o 6 vecinos”.

Esto se puede generalizar aún más a reglas no totalistas, que consideran las posiciones de las células vivas en el vecindario de una célula dada en lugar de simplemente el número de ellas cuando se determina el nacimiento y la supervivencia. Algunas de las variaciones no totalistas conocidas de la Vida son la vida y los *tlife* y *Snowflakes*, y el primero tiene muchas variantes de interés. Existen variaciones adicionales en la Vida, aunque la gran mayoría de estos universos son demasiado caóticos o desolados como para justificar una exploración extensa.

Otros tipos populares de autómatas celulares incluyen reglas de generaciones, como *Brian’s Brain*, en el cual las células “envejecen” con el paso del tiempo en lugar de morir de inmediato.

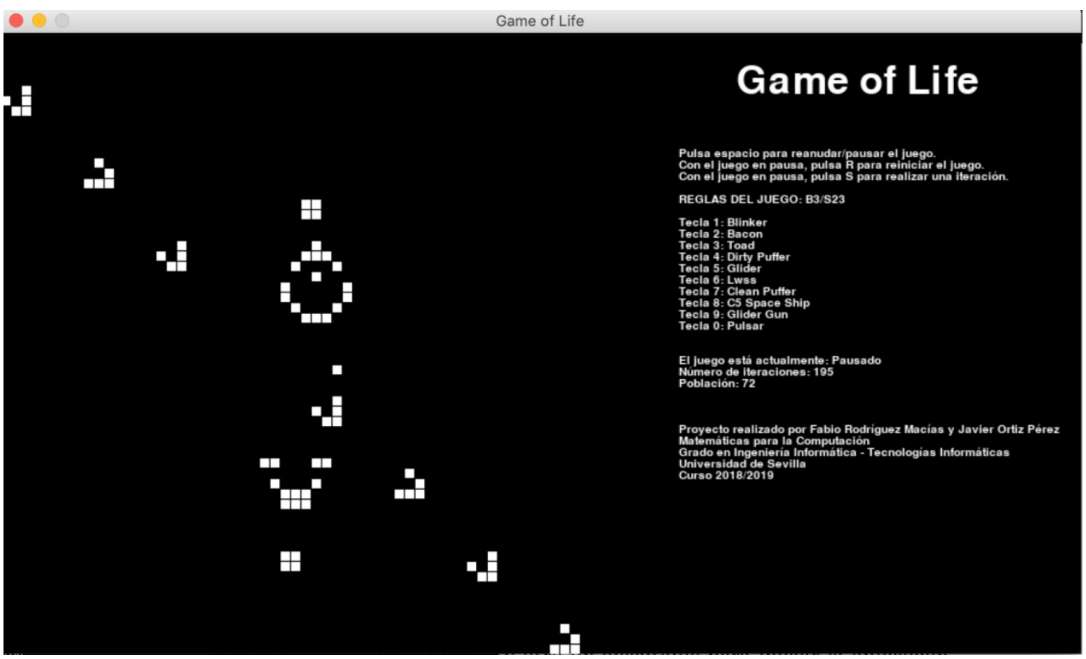
Las reglas de Conway también pueden generalizarse para que en lugar de dos estados (vivo y muerto) haya tres o más. Las transiciones de estado se determinan mediante un sistema de ponderación o una tabla que especifica reglas de transición separadas para cada estado; por ejemplo, las familias de reglas de "Tabla de reglas" y "Vida ponderada" de Mirek Cellebration incluyen reglas de muestra equivalentes a la vida de Conway.

Los patrones relacionados con los fractales y los sistemas fractales también se pueden observar en ciertas variaciones similares a la Vida. Por ejemplo, el autómata 12/1 genera cuatro aproximaciones muy cercanas al triángulo de Sierpiński cuando se aplica a una sola célula viva.

1.8 Implementación de la aplicación

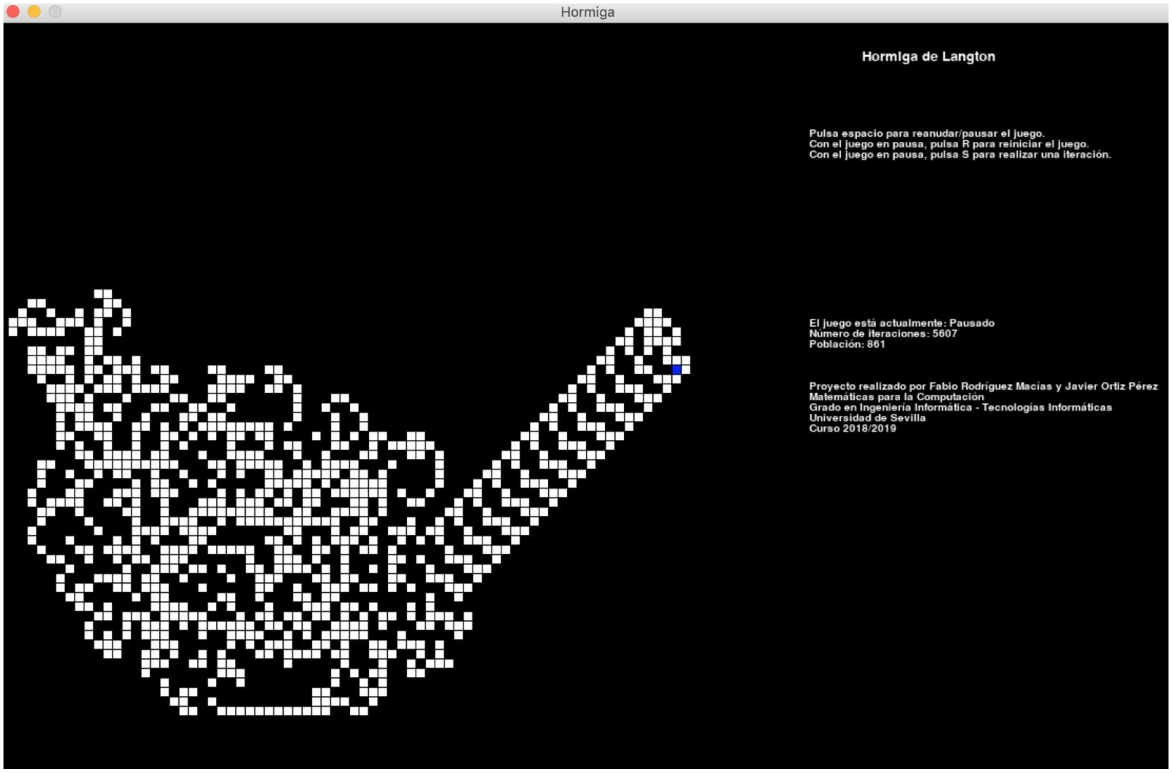
Hemos realizado una implementación del juego de la vida con *Python* y la librería *Pygame*. Cuando ejecute el juego debe introducir en la consola diversos parámetros que corresponderán al tamaño del tablero (matriz cuadrada NxN), el número de vecinos necesarios para que una célula nazca, menor numero para que siga viviendo y no muera por soledad, y mayor numero para que siga viviendo y no muera por sobrepoblación.

Esta es la interfaz del programa:



En el menú de opciones (situado a la izquierda del tablero interactivo) tenemos información sobre los diferentes comandos que le podemos introducir al Juego de la Vida, además de información sobre el estado del juego (Pausado o En ejecución), el número de iteraciones o *ticks*, y el número de células vivas (Población). Para reiniciar el juego o introducir ejemplos es necesario que esté pausado.

También hemos realizado la implementación de otro Autómata Celular llamado la Hormiga de Langton. En este caso el algoritmo no precisa de parámetros de entrada:



De forma análoga al Juego de la Vida tenemos un menú de información donde podemos ver el estado actual (En Ejecución o Pausado), el número de iteraciones que ha realizado y la población en la última generación. Para reiniciar el juego es necesario que esté pausado.

Bibliografía

David Alejandro Reyes Gómez - Descripción y Aplicaciones de los Autómatas Celulares

Autómata Celular - Wikipedia

Sistema dinámico - Wikipedia

Fernando Sánchez Caparrini – Autómatas Celulares

Fractales - Wikipedia

"Exposure" - *The Life Lexicon*. Stephen.

Nehaniv, Chrystopher L. (2002) - *Self-Reproduction in Asynchronous Cellular Automata*.

Cellular Automata FAQ – Conway's Game of Life

Conway's Game of Life – *Wikipedia*

Conway Life - Wiki